**Final Project – Part 2**

**1.)**

**Conceptual Diagram/Schema for database**

A black and white screen

Description automatically generated

The Schema diagram has 5 tables which are connected based on single table apps\_dim (Star Schema design). The tables have been normalized wherein the primary table is apps\_dim and the others are joined based on the corresponding keys as explained below -

**Table1: apps\_dim**

Entities:

* app\_id (Primary Key)
* app (VARCHAR)
* size (DOUBLE)
* installs (INTEGER)
* last\_updated (VARCHAR)
* current\_ver (VARCHAR)
* android\_ver (VARCHAR)

Relations: No explicit foreign keys

**Table2: app\_categories\_genres**

Entities:

* category\_id (Primary Key)
* app\_id (Foreign Key referencing apps\_dim)
* category (VARCHAR)
* app\_type (VARCHAR)
* content\_rating (VARCHAR)
* genres (VARCHAR)

Relations:

app\_categories\_genres have Many-to-one relationship with apps\_dim table through app\_id as foreign key.

**Table3: app\_ratings**

Entities:

* rating\_id (Primary Key)
* app\_id (Foreign Key referencing apps\_dim)
* rating (DOUBLE with a CHECK constraint)
* reviews (INT)

Relations:

app\_ratings has one-to-one relationship with apps\_dim table through app\_id as foreign key.

**Table4: app\_price**

Entities:

* price\_id (Primary Key)
* app\_id (Foreign Key referencing apps\_dim)
* price (DOUBLE)

Relations: app\_ price has one-to-one relationship with apps\_dim table through app\_id as foreign key.

**Table5: app\_reviews**

Entities:

* review\_id (Primary Key)
* app\_id (Foreign Key referencing apps\_dim)
* review (TEXT)
* sentiment (TEXT)

Relations: app\_ reviews have Many-to-one relationship with apps\_dim table through app\_id as foreign key. (**the join will be left join with apps\_dim table, since not all apps will have reviews submitted)**

**2.)**

**Database Constraints**:

1. apps\_dim:

* all the fields in this table has NOT NULL constraint.
* App\_id is PRIMARY KEY and AUTO\_INCREMENT is assigned to initialize with a value for each entry in table.

1. app\_categories\_genres:

* all the fields in this table has NOT NULL constraint.
* Category\_id is PRIMARY KEY and AUTO\_INCREMENT is assigned to initialize with a value for each entry in table
* App\_id is FOREIGN KEY referencing apps\_dim(app\_id) table

1. app\_ratings:

* all the fields in this table has NOT NULL constraint.
* rating\_id is PRIMARY KEY and AUTO\_INCREMENT is assigned to initialize with a value for each entry in table.
* App\_id is FOREIGN KEY referencing apps\_dim(app\_id) table
* rating has CHECK constraint which ensures that only values between 1.0 and 5.0(inclusive) are present in the data

1. app\_price:

* all the fields in this table has NOT NULL constraint.
* price\_id is PRIMARY KEY and AUTO\_INCREMENT is assigned to initialize with a value for each entry in table.
* App\_id is FOREIGN KEY referencing apps\_dim(app\_id) table

1. app\_reviews:

* NOT NULL constraint is not enforced in this since, not every app is mandated to have a review
* review\_id is PRIMARY KEY and AUTO\_INCREMENT is assigned to initialize with a value for each entry in table.
* App\_id is FOREIGN KEY referencing apps\_dim(app\_id) table

**3.)**

**Code**: Only build queries result screenshot posted here, the whole code along with build queries in final.sql file

**Build Query results:**

**Query1:**

**A screenshot of a computer

Description automatically generated**

**Query2:**

**A screenshot of a computer

Description automatically generated**

**Query3:**

**A screenshot of a computer

Description automatically generated**

**Query4:**

**A screenshot of a computer

Description automatically generated**

**Query5:**

**A screenshot of a computer

Description automatically generated**

**Query6:**

**A screenshot of a computer

Description automatically generated**

**Query7:**

**A screenshot of a computer

Description automatically generated**

**4.) Overall Contribution:**

|  |  |  |
| --- | --- | --- |
| Name | Task | Contribution |
| Prashul Kumar | **Initial data cleaning,**  **Conceptual Schema,**  **Database,**  **Code** | Worked on developing the conceptual schema design, normalization of table, cleaning data before loading into the database (such as handling NA and removing special characters in Python),  ideating the parameters and results to analyze thereby developing some queries for results |
| Jagadeesh | **Conceptual Schema, Database,**  **Code** | Worked on developing and drawing the ERD diagram and database normalization (part of schema),  Also worked on designing the database relations (joins) and constraints on each table,  Inserting data into the individual tables and some queries for result |
| Dikshak | **Code,**  **Word Document** | Worked on building the queries which we are planning to visualize or analyze in the front-end and designing this document |